

NASA/CR-2001-210853  
ICASE Report No. 2001-7



## **Tactical Conflict Detection and Resolution in a 3-D Airspace**

*Gilles Dowek  
INRIA, Le Chesnay Cedex, France*

*César Muñoz and Alfons Geser  
ICASE, Hampton, Virginia*

*ICASE  
NASA Langley Research Center  
Hampton, Virginia*

*Operated by Universities Space Research Association*



National Aeronautics and  
Space Administration

Langley Research Center  
Hampton, Virginia 23681-2199

Prepared for Langley Research Center  
under Contract NAS1-97046

April 2001

<b>Report Date</b> <i>("DD MON YYYY")</i> 00APR2001	<b>Report Type</b> N/A	<b>Dates Covered (from... to)</b> <i>("DD MON YYYY")</i>
<b>Title and Subtitle</b> Tactical Conflict Detection and Resolution in a 3-D Airspace		<b>Contract or Grant Number</b>
		<b>Program Element Number</b>
<b>Authors</b> Gilles Dowek César Muñoz and Alfons Geserv		<b>Project Number</b>
		<b>Task Number</b>
		<b>Work Unit Number</b>
<b>Performing Organization Name(s) and Address(es)</b> ICASE NASA Langley Research Center Hampton, Virginia		<b>Performing Organization Number(s)</b> ICASE Report No. 2001-7
<b>Sponsoring/Monitoring Agency Name(s) and Address(es)</b>		<b>Monitoring Agency Acronym</b>
		<b>Monitoring Agency Report Number(s)</b>
<b>Distribution/Availability Statement</b> Approved for public release, distribution unlimited		
<b>Supplementary Notes</b>		
<b>Abstract</b> Abstract. This paper presents an algorithm for detection and resolution of air traffic conflicts in a 3-dimensional (3-D) airspace for two aircraft, namely ownship and intruder. A conflict is a projected incursion of the intruder aircraft within the protected zone of the ownship. A solution is a single maneuver, to be performed by the ownship, that effectively keeps the required minimum separation without cooperation of the intruder aircraft. The input to the algorithm is the state information, i.e., horizontal position, altitude, ground track, and vertical and ground speed, of both aircraft. The algorithm outputs a set of solutions. Each solution modifies only one state parameter of the ownship: ground track, ground speed, or vertical speed. The proposed algorithm is suitable for formal verification.		
<b>Subject Terms</b>		
<b>Document Classification</b> unclassified	<b>Classification of SF298</b> unclassified	
<b>Classification of Abstract</b> unclassified	<b>Limitation of Abstract</b> unlimited	
<b>Number of Pages</b> 16		

# TACTICAL CONFLICT DETECTION AND RESOLUTION IN A 3-D AIRSPACE \*

GILLES DOWEK<sup>†</sup>, CÉSAR MUÑOZ<sup>‡</sup>, AND ALFONS GESER<sup>§</sup>

**Abstract.** This paper presents an algorithm for detection and resolution of air traffic conflicts in a 3-dimensional (3-D) airspace for two aircraft, namely ownship and intruder. A conflict is a projected incursion of the intruder aircraft within the protected zone of the ownship. A solution is a single maneuver, to be performed by the ownship, that effectively keeps the required minimum separation without cooperation of the intruder aircraft. The input to the algorithm is the state information, i.e., horizontal position, altitude, ground track, and vertical and ground speed, of both aircraft. The algorithm outputs a set of solutions. Each solution modifies only one state parameter of the ownship: ground track, ground speed, or vertical speed. The proposed algorithm is suitable for formal verification.

**Key words.** conflict detection, conflict resolution, CD&R algorithm, 3-dimensional, collision avoidance

**Subject classification.** Computer Science

**1. Introduction.** One of the main elements of the Free-Flight concept [11] is the redistribution of responsibilities for air traffic separation. Under Free-Flight rules, each aircraft with an appropriate level of equipment is responsible to assure separation with other aircraft in the vicinity. To support this mode of operation, several automated decision support systems are being proposed. In this context, Conflict Detection and Resolution algorithms (CD&R) are designed to warn pilots about an imminent loss of separation, and to assist them in a corrective maneuver.

In this paper, we present a *tactical* CD&R algorithm in a 3-D space for *two* aircraft. In CD&R-related literature, *tactical* algorithms use only state information to project aircraft trajectories. Due to this intentionally limited source of information, they are intended to be used with short lookahead times (a few minutes, typically 5-10) during which aircraft are supposed to follow straight flight paths. *Strategic* approaches, in contrast, use intent information such as flight plans, and uncertainties such as weather conditions. They may have lookahead windows of several minutes and even hours. For a survey on CD&R methods see [7].

The input to our algorithm is the state information, i.e., 3-D position, ground track, vertical speed, and ground speed, of two aircraft. We distinguish one aircraft as the *ownship* and the other as the *intruder*. Loss of separation between the aircraft is predicted via linear projections on time of the state parameters. In case of a predicted conflict, the algorithm proposes several solutions for the ownship. Every solution is a single maneuver that effectively avoids the conflict. The maneuver modifies only one parameter of the ownship. This constraint produces finitely many solutions, simplifies the calculations performed by the algorithm, and is simple to conceive and to perform by the crew.

In Section 2, we briefly survey some methods for conflict detection and resolution that inspired our approach. In Section 3, we present our approach and its theoretical support. In Section 4, we give an algorithm to find convenient 3-D conflict solutions. A prototype implementation is described in Section 5. The last section summarizes our work and suggests lines of research for future work.

---

\*This work was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-97046 while the authors were in residence at ICASE, NASA Langley Research Center, Hampton, VA 23681-2199, USA.

<sup>†</sup>INRIA, Domaine de Voluceau - Rocquencourt - B.P. 105, 78153 Le Chesnay Cedex, France, e-mail: Gilles.Dowek@inria.fr.

<sup>‡</sup>ICASE, Mail Stop 132C, NASA Langley Research Center, Hampton, VA 23681-2199, USA, e-mail: munoz@icase.edu.

<sup>§</sup>ICASE, Mail Stop 132C, NASA Langley Research Center, Hampton, VA 23681-2199, USA, e-mail: geser@icase.edu.

**2. Conflict Detection and Resolution Supporting Free Flight.** Distributed Air/Ground Traffic Management (DAG-TM) [1] is a set of conceptual elements, developed within the Advanced Air Transportation Technologies project at NASA, that defines modes of operation supporting the Free-Flight concept.

Safety assessment of new air traffic management systems is a main issue in DAG-TM. Prototype tools such as the Autonomous Operations Planner (AOP) are being developed at NASA Langley to study the feasibility of self separation. Systems with similar goals have been proposed in other research laboratories, e.g., the Future ATM Concepts Evaluation Tool (FACET) [3] at NASA Ames<sup>1</sup> and the Airborne Separation Assurance System (ASAS) [6] at the National Aerospace Laboratory (NLR) in the Netherlands. All these tools implement CD&R algorithms.

Standard safety assessment techniques such as testing and simulation, although useful, have serious limitations in new systems which are significantly more autonomous than the older ones. Given the critical nature of the problem, we believe that safety statements should be made and verified formally, and that proofs should be checked by machine.

With the above premise in mind, we have examined at the AOP conflict detection and resolution procedures. The AOP will eventually provide both tactical and strategic CD&R resolution. In what follows, we consider the conflict detection part of the CD&R AOP algorithm, which is an adaptation of a deterministic procedure implemented within the ground NASA's Center/TRANCON Automation System (CTAS) [12].<sup>2</sup>

**2.1. Conflicts and Protected Zone.** Two aircraft are said to be in *conflict* if their vertical separation is (strictly) less than  $H$ ,  $H > 0$ , and their horizontal separation is (strictly) less than  $D$ ,  $D > 0$ . A body in the 3-D space, called *protected area*, is assigned to each aircraft such that a conflict is equivalent to an intrusion of another airplane into its protected area. The protected area forms a cylinder (hockey-puck, nickel, pizza) of altitude  $H$  and radius  $D$  around the position of the aircraft. The values  $H = 1000$  ft (feet) and  $D = 5$  nm (nautical miles) are commonly used.

Note that the boundaries are not considered part of the protected area. We will see later that this choice enables optimal ownship maneuvers that touch the boundaries of the intruder protected area.

**2.2. AOP Conflict Detection.** The input to the strategic AOP conflict detection algorithm [8] is a set of trajectories, one of which is the ownship trajectory. A *trajectory* is a list of points, called *nodal points*, which are assumed to be joined by linear segments. Each nodal point contains the *intended* aircraft state (position and velocity vector) at a given time. Since conflict detection is assumed to be asynchronous, the first step of the algorithm is to synchronize all the trajectories. This is done by taking time steps of duration  $\Delta$  and then measuring the distance between the trajectories at every step during a lookahead period of time. If after  $n$  time steps there is a violation of the ownship protected zone, then a conflict is detected. We illustrate the situation in Figure 2.1, where a loss of separation occurs at time  $n\Delta$ . The algorithm implements several heuristics to avoid unnecessary calculations.

Since the algorithm does not compute the actual time when the first loss of separation occurs, the choice of  $\Delta$  is crucial in this approach. Indeed, if  $\Delta$  is too large, near misses can occur without being detected. CTAS uses  $\Delta = 10$  seconds.

The synchronization step introduces, in fact, a time and space discretization. In recent work, we have discovered that discretization makes formal verification more difficult [4]. Moreover, discretization of trajectories can lead to an accumulation of modeling inaccuracies that lead to imprecise conclusions. We

<sup>1</sup>FACET is a CD&R analysis tool rather than a flight deck decision support tool.

<sup>2</sup>The AOP detection algorithm also detects conflicts with hazard areas. That kind of detection is outside the scope of this paper.

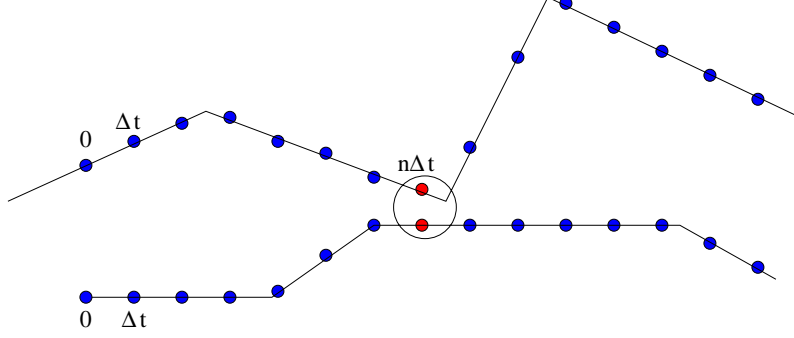


FIG. 2.1. *Strategic AOP conflict detection*

have successfully verified a conflict alerting algorithm using a continuous trajectory model [9]. Continuous trajectory models are well suited to formal verification.

The approaches exhibited in the remainder of this section use a 2-D geometry.

**2.3. Conflict Detection.** Let  $\vec{a}_o$ ,  $\vec{v}_o$  and  $\vec{a}_i$ ,  $\vec{v}_i$  be the position and ground speed vector of the ownship and intruder aircraft, respectively, at time 0. We assume that the ground speeds are constants. Separation is lost at time  $t$  if and only if the projected distance between both aircraft at time  $t$  is strictly less than  $D$ , i.e.,

$$|(\vec{a}_o - \vec{a}_i) + t(\vec{v}_o - \vec{v}_i)|^2 < D^2 . \quad (2.1)$$

This constraint has solutions for  $t$  if and only if Equation 2.2

$$|(\vec{a}_o - \vec{a}_i) + t(\vec{v}_o - \vec{v}_i)|^2 = D^2 \quad (2.2)$$

has two solutions  $t_1$  and  $t_2$  such that  $t_1 \neq t_2$ . The two solutions correspond to the times when the loss of separation starts and ends. Note that by definition, if  $t_1 = t_2$  no loss of separation occurs.

The above procedure is used in both ASAS (modified potential field) and FACET (Bilimoria's geometric optimization) algorithms for conflict detection in 2-D [6, 2].

**2.4. Modified Potential Field Resolution.** If a conflict is predicted, the modified potential field approach (originally due to [5] and implemented in ASAS) computes the time of closest separation  $\tau$  between the ownship and the aircraft ( $\tau = (t_1 + t_2)/2$ ). Then, a new speed vector for the ownship is calculated such that the distance at time  $\tau$  between the aircraft is exactly  $D$ . The solution is illustrated in Figure 2.2 where  $\vec{a}_o(\tau)$  and  $\vec{a}_i(\tau)$  are the ownship and intruder projected positions, respectively, at time  $\tau$ ,  $\vec{v}'_o$  is the ownship's new speed vector, and  $\vec{a}'_o(\tau)$  is the new projected position of the ownship at time  $\tau$ .

The modified potential field approach does not solve conflicts. If the ownship maneuvers towards the new speed vector, and no further action is taken, then there will be a conflict after time  $\tau$ . Without cooperation of the intruder, the ownship will have to pursue repeated maneuvers to only approximate a solution. To completely solve the conflict, even an *infinite* number of maneuvers are necessary.

**2.5. Geometric Optimization Resolution.** In this approach, proposed by Bilimoria [2] and implemented in FACET, the intruder is considered fixed in space, and the ownship position  $\vec{a}$  and velocity vector  $\vec{v}$  are taken relative to the intruder state, i.e.,  $\vec{a} = \vec{a}_o - \vec{a}_i$  and  $\vec{v} = \vec{v}_o - \vec{v}_i$ . A new relative speed vector for the ownship solves the conflict if it does not intersect the interior of the intruder protected area. Among

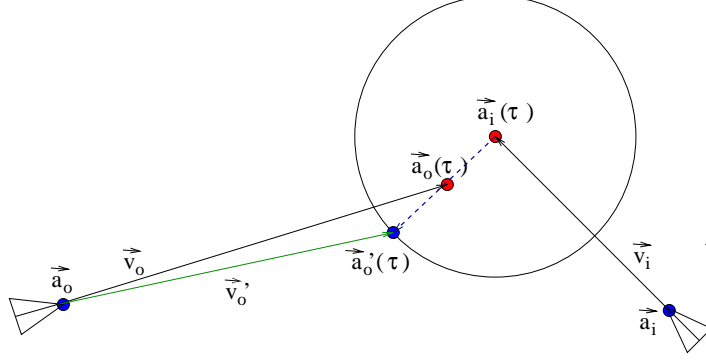


FIG. 2.2. *Modified Potential Field resolution*

the infinitely many new speed vectors that solve the conflict, Bilimoria chooses those which minimize their angle to the original speed vector. Such speed vectors are called *optimal*.

As illustrated in Figure 2.3, any optimal solution is tangential to the intruder's protected zone. Any other solution requires a greater change of the ownship relative ground track. Each touch point  $\vec{A}$  and  $\vec{B}$  determines a new ownship relative ground track which is optimal under certain constraints. For instance the target point  $\vec{A}$  is optimal under the constraint that only a left turn may be made. The length of the speed vector may be arbitrarily chosen. A minimal solution is proposed in [2].

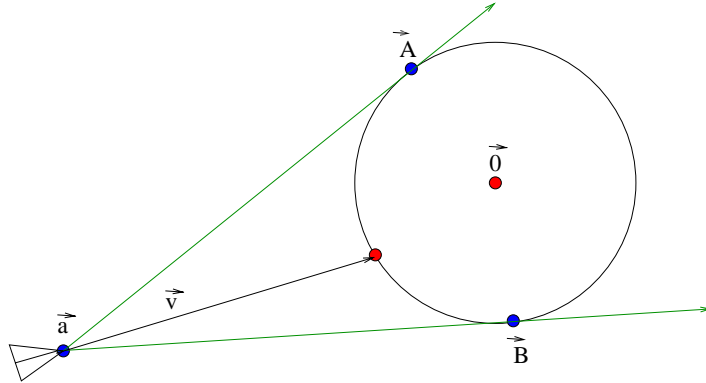


FIG. 2.3. *Geometric Optimization resolution*

**3. An Approach to 3-D CD&R.** The Geometric Optimization algorithm in [2] uses a 2-D geometry, i.e., it detects and solves conflicts in the horizontal plane. In the Modified Potential Field algorithm implemented in ASAS [6], 3-D conflicts are decomposed into horizontal and vertical conflicts which are detected and solved independently. Then, the solutions are composed to obtain a 3-D maneuver. This approach is appealing for its simplicity, but it is rather difficult to prove correct.

We pursue a true 3-D geometric analysis to conflict detection and resolution. In this section we extend Bilimoria's horizontal CD&R approach to three dimensions.

Given the positions and speed vectors of two aircraft (ownship and intruder), we compute the *relative position*  $\vec{a} = \vec{v}_o - \vec{v}_i$  and *relative speed*  $\vec{v} = \vec{v}_o - \vec{v}_i$  of the ownship with respect to intruder referential. We take a coordinate system where the origin is at the intruder position and  $a_x < 0, a_y = 0$  (see Figure 3.1).

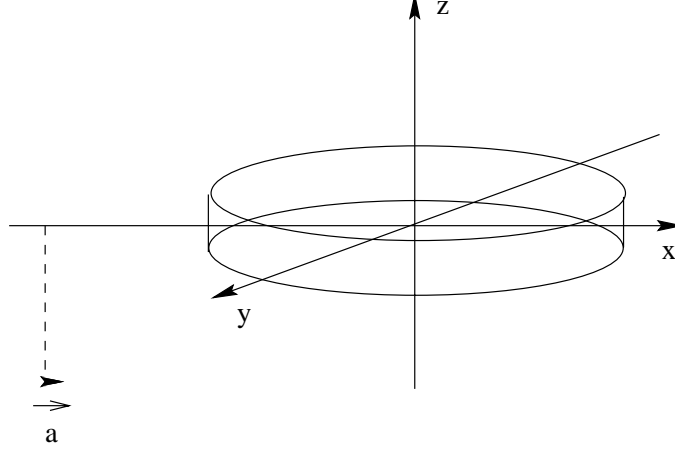


FIG. 3.1. 3-D coordinate system and protected zone

**3.1. 3-D Conflict Detection.** The *relative ownship trajectory*  $L$  is the half line

$$L = \{\vec{a} + t\vec{v} \mid t > 0\}. \quad (3.1)$$

We say that there is a *conflict* between the ownship and the intruder aircraft if and only if  $\vec{a} \in P$ , where  $P$  is the protected zone

$$P = \{(x, y, z) \mid x^2 + y^2 < D^2 \text{ and } -H < z < H\}. \quad (3.2)$$

The two aircraft are *predicted* to be in *conflict* if the relative trajectory  $L$  intersects the protected zone  $P$ . To compute the time interval where the projected intrusion occurs, we compute the intersection between  $L$  and the boundary of  $P$ . The boundary consists of two parts: (1) the lateral surface around the cylinder

$$P_1 = \{(x, y, z) \mid x^2 + y^2 = D^2 \text{ and } -H \leq z \leq H\} \quad (3.3)$$

and (2) the top and bottom bases

$$P_2 = \{(x, y, z) \mid x^2 + y^2 < D^2 \text{ and } |z| = H\}. \quad (3.4)$$

Observe that we count the top and bottom circles to the lateral surface and not to the bases.

If the half line  $L$  does not intersect  $P_1$  nor  $P_2$ , then the two aircraft are predicted to be in conflict. Otherwise, we compute times  $t_1$  and  $t_2$  such that aircraft will be in conflict during the interval  $t_1 < t < t_2$ .

Assume that the ownship and the intruder are not in conflict. The following algorithm returns *true* if the two aircraft are predicted to be in conflict, and *false* otherwise. It has two parameters:  $\vec{a} = (a_x, 0, a_z)$  and  $\vec{v} = (v_x, v_y, v_z)$ .

1. Case  $v_z = 0$ . This case corresponds to 2-D resolution. Return *true* if  $|a_z| \leq H$ ,  $v_x > 0$ , and the discriminant  $D^2(v_x^2 + v_y^2) - a_x^2 v_y^2$  is positive. Otherwise, return *false*.
2. Case  $v_z \neq 0$ . Let  $t_1 = \frac{H - a_z}{v_z}$  and  $t_2 = \frac{-H - a_z}{v_z}$ . These are the times when the trajectory reaches the altitudes  $\pm H$ . Let  $d_1, d_2$  denote the squares of the horizontal distances to the intruder at times  $t_1, t_2$ , respectively. We have three cases:
  - (a) Case  $d_1 < D^2$  and  $d_2 < D^2$ . There are two intersections with the bases  $P_2$ , provided that the times  $t_1, t_2$  are in the future. Return *true* if  $\min(t_1, t_2) \geq 0$ , and return *false* otherwise.

- (b) Case  $d_1 \geq D^2$  and  $d_2 \geq D^2$ . In this case there are no intersections with the bases, but there may be intersections with the lateral surface  $P_1$ . Return *true* if  $v_x > 0$  and  $D^2(v_x^2 + v_y^2) - a_x^2 v_x^2 > 0$ . Otherwise, return *false*.
- (c) Case  $d_1 \leq D^2$  and  $d_2 > D^2$ . We have one intersection with a base at time  $t_1$ . Return *true* if  $t_1 \geq 0$ . Otherwise, return *false*.
- (d) Case  $d_1 > D^2$  and  $d_2 \leq D^2$ . We have one intersection with a base at time  $t_2$ . Return *true* if  $t_2 \geq 0$ . Otherwise, return *false*.

**3.2. 3-D Conflict Resolution: The Target Set.** We assume that the ownship and the intruder are not yet in conflict, but that they are predicted to be in a conflict. We want to modify the relative speed vector  $\vec{v}$  to a vector  $\vec{v}'$  such that the half line  $L' = \{\vec{a} + \vec{v}'t \mid t > 0\}$ , does not intersect the protected zone.

Among the various solutions, we focus on those that modify the vector  $\vec{v}'$  in an optimal way, i.e., such that  $L'$  touches the boundary of  $P$ . We will also consider the special solution  $\vec{v}' = 0$  (i.e., the two aircraft have exactly the same speed) as optimal. Positive multiples of optimal solutions are again optimal solutions. Hence, we shall first characterize the *directions* of optimal solutions.

In what follows we assume that  $\vec{a}$  is not on the boundary of the protected area  $P$  (the case where  $\vec{a}$  is on this boundary must be handled separately). direction of an optimal solution is determined by one target point of the half line  $L'$  with the boundary of  $P$ . We must indeed be aware of half lines touching one of the boundary circles of the lateral surface in two points if  $\vec{a}_z = \pm H$ . Let a *target set* of  $\vec{a}$  be a set of target points such that for every optimal solution one of its target points is in the set. The kernel of our 3-D conflict resolution algorithm is the computation of a suitable target set.

A target point may be either on the lateral surface  $P_1$  or on the bases  $P_2$ . If we have a target point on one of the bases, then  $L'$  must touch an open line segment of the base, and moreover it must touch the lateral surface at two points. In this case we decide to take one of the latter points as target points. So all target points are on the lateral surface.

A target point satisfies two things: (1) it is on the boundary of the protected zone, and (2) the half line  $L'$  passing on this point must not intersect the protected zone (only its boundary). If the target point has coordinates  $(x, y, z)$  the first condition rephrases  $x^2 + y^2 = D^2$  and  $-H \leq z \leq H$ . The second is that for all time  $t > 0$ ,  $(a_x + tv_x, tv_y, a_z + tv_z) \notin P$ .

Our main goal on this computation is to remove the phrase “for all time  $t > 0$ ” to get an algebraic characterization of the target set. Let  $t_0$  be the time when the half line  $L'$  crosses the target point  $(x, y, z)$ , i.e.,  $x = a_x + v'_x t_0, y = v'_y t_0, z = a_z + v'_z t_0$ . We define  $T = t/t_0$  as the normalized time on this half line. The half line  $L'$  can be written as the set of points  $(a_x + T(x - a_x), Ty, a_z + T(z - a_z))$  for  $T > 0$ . The condition that this half line does not intersect the protected zone is rephrased: For all  $T > 0$  either

$$(a_x + T(x - a_x))^2 + T^2 y^2 \geq D^2 \quad (3.5)$$

or

$$|a_z + T(z - a_z)| \geq H. \quad (3.6)$$

Using  $x^2 + y^2 = D^2$ , Formula 3.5 yields

$$(T - 1)[(T - 1)((x - a_x)^2 + y^2) + 2(D^2 - a_x x)] \geq 0 \quad (3.7)$$

and Formula 3.6 yields

$$|z + (T - 1)(z - a_z)| \geq H. \quad (3.8)$$



First we consider the target points such that  $-H < z < H$  and then those such that  $|z| = H$ .

- Case  $-H < z < H$ . For some time  $T$  in an interval around 1, Formula 3.8 does not hold, i.e.,

$$|z + (T - 1)(z - a_z)| < H.$$

Therefore, Formula 3.7 must hold for an interval around 1. Thus, we must have  $D^2 - a_x x = 0$  and hence

$$x = D^2/a_x.$$

As  $x^2 + y^2 = D^2$ , we get

$$y = \varepsilon \sqrt{D^2 - (D^2/a_x)^2} = -\varepsilon D \sqrt{a_x^2 - D^2}/a_x,$$

where  $\varepsilon$  is  $\pm 1$ .

As the target set must be on the boundary of the protected area, we must have  $-a_x \geq D$ .

It is easy to check that all points such that  $x = D^2/a_x$ ,  $y = -\varepsilon D \sqrt{a_x^2 - D^2}/a_x$  and  $-H < z < H$  are target points.

- Case  $|z| = H$ . If  $z$  and  $z - a_z$  have the same sign, Formula 3.8 is equivalent to  $T \geq 1$ , and we must have for all  $T < 1$

$$(T - 1)[(T - 1)((x - a_x)^2 + y^2) + 2(D^2 - a_x x)] \geq 0$$

and hence  $D^2 - a_x x < 0$ , i.e.,  $x < D^2/a_x$ . Symmetrically, if  $z$  and  $z - a_z$  have different signs, we must have  $x > D^2/a_x$ .

The relative signs of  $z$  and  $z - a_z$  depends of  $z$ , which can be  $\pm H$ , and the position of  $a_z$  with respect to  $\pm H$ .

We analyze all the cases.

1. Case  $-H < a_z < H$  and  $-a_x > D$ . In this case, the target set is the set of points  $(x, y, z)$  of  $P_1$  such that  $x = D^2/a_x$  or  $(z = H \text{ and } x < D^2/a_x)$  or  $(z = -H \text{ and } x < D^2/a_x)$ . See Figure 3.2.

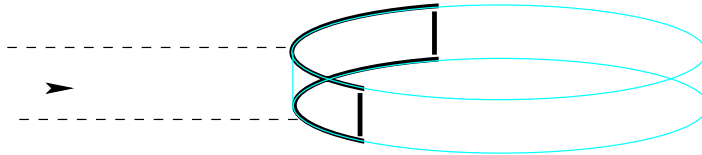


FIG. 3.2. Target set. Case  $-H < a_z < H$  and  $-a_x > D$

2. Case  $a_z < -H$  and  $-a_x \geq D$ . In this case, the target set is the set of points  $(x, y, z)$  of  $P_1$  such that  $x = D^2/a_x$  or  $(z = H \text{ and } x < D^2/a_x)$  or  $(z = -H \text{ and } x > D^2/a_x)$ . See Figure 3.3.
3. Case  $H < a_z$  and  $-a_x \geq D$ . In this case, the target set is the set of points  $(x, y, z)$  of  $P_1$  such that  $x = D^2/a_x$  or  $(z = H \text{ and } x > D^2/a_x)$  or  $(z = -H \text{ and } x < D^2/a_x)$ . See Figure 3.4.
4. Case  $a_z < -H$  and  $-a_x < D$ . In this case, the target set is the set of points  $(x, y, z)$  of  $P_1$  such that  $z = -H$ . See Figure 3.5.
5. Case  $H < a_z$  and  $-a_x < D$ . In this case, the target set is the set of points  $(x, y, z)$  of  $P_1$  such that  $z = H$ . See Figure 3.6.

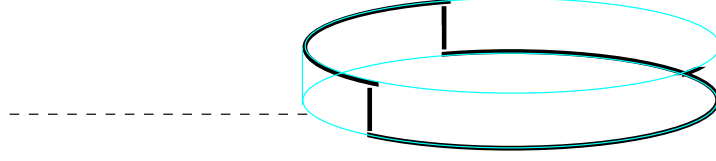


FIG. 3.3. *Target set. Case  $a_z < -H$  and  $-a_x \geq D$*

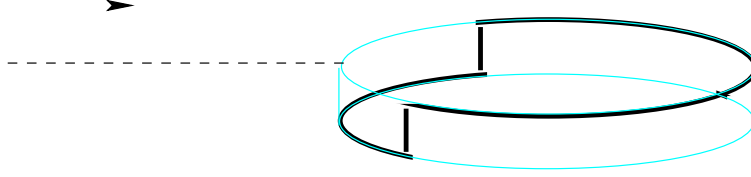


FIG. 3.4. *Target set. Case  $H < a_z$  and  $-a_x \geq D$*

6. Case  $a_z = -H$  and  $-a_x > D$ . In this case, the target set is the set of points  $(x, y, z)$  of  $P_1$  such that  $x = D^2/a_x$  or  $(z = H$  and  $x < D^2/a_x)$  or  $z = -H$ . See Figure 3.7. Notice that we get the same set of directions as the target set in Figure 3.3. These directions are already included in case (2).
7. Case  $a_z = H$  and  $-a_x > D$ . In this case, the target set is the set of points  $(x, y, z)$  of  $P_1$  such that  $x = D^2/a_x$  or  $z = H$  or  $(z = -H$  and  $x < D^2/a_x)$ . See Figure 3.8. We get the same set of directions as the target set in Figure 3.4. These directions are already included in case (3).

**4. Constrained Solutions.** In the previous section, we got an infinite set of target points (which defines optimal directions for  $\vec{v}'$ ). A new ownship speed vector  $\vec{v}'_o$  can be calculated from any point in the target set and any length of the relative speed vector  $\vec{v}'$ . Among all these vectors, representing maneuvers, some are more convenient than others. In this section, we select solutions where only one parameter of the (absolute) ownship speed vector is modified, i.e., ground speed, ground track, or vertical speed.

As we have seen, the target set is a subset of the set of points in the set  $P' = P'_1 \cup P'_2$  (see Figure 4.1), where

$$\begin{aligned} P'_1 &= \{(x, y, z) \mid x^2 + y^2 = D^2 \text{ and } z = \varepsilon H \text{ and } \varepsilon = \pm 1\} && \text{(points in circles),} \\ P'_2 &= \{(x, y, z) \mid x = D^2/a_x \text{ and } y = -\varepsilon D \sqrt{a_x^2 - D^2}/a_x \text{ and } \varepsilon = \pm 1\} && \text{(points in lines).} \end{aligned}$$

To get constrained solutions, we first compute the solutions on this set satisfying a given constraint. For that, we assume that at some time  $t$ , the point  $(x, y, z) = (a_x + t(v'_{ox} - v_{ix}), t(v'_{oy} - v_{iy}), a_z + t(v'_{oz} - v_{iz}))$  is in  $P'$  and we proceed by case analysis on  $P'_1$  and  $P'_2$ . A special case, when  $\vec{v}' = 0$ , is also considered. We ignore points where  $t > 0$ . Once the solutions are found, we check whether they belong to the target set or not.

**4.1. Ground Speed Change Only.** We have  $\vec{v}'_o = (k v_{ox}, k v_{oy}, v_{oz})$  for some  $k > 0$ . We must determine the possible  $k$  positive such that at some time  $t$ , the point  $(x, y, z) = (a_x + t(v'_{ox} - v_{ix}), t(v'_{oy} - v_{iy}), a_z + t(v'_{oz} - v_{iz}))$  is on the target set.

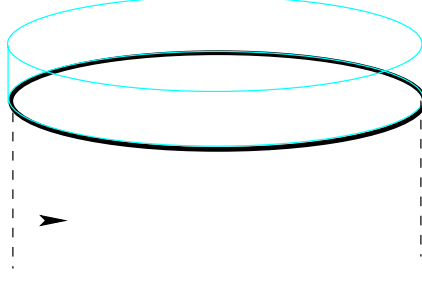


FIG. 3.5. *Target set. Case  $a_z < -H$  and  $-a_x < D$*

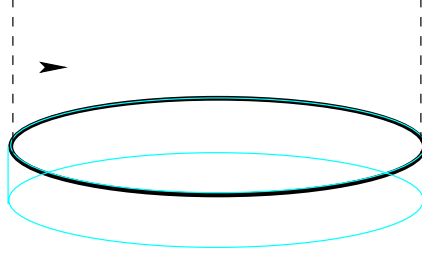


FIG. 3.6. *Target set. Case  $H < a_z$  and  $-a_x < D$*

1. Points on the circles  $P'_1$ . We have

$$x^2 + y^2 = D^2 \quad (4.1)$$

$$z = \varepsilon H, \quad (4.2)$$

where  $\varepsilon = \pm 1$ . Equation 4.2 yields

$$t = (\varepsilon H - a_z) / (v_{oz} - v_{iz}).$$

Equation 4.1 rewrites to

$$(a_x + t(kv_{ox} - v_{ix}))^2 + (t(kv_{oy} - v_{iy}))^2 = D^2,$$

i.e.,

$$t^2(v_{ox}^2 + v_{oy}^2)]k^2 + [2(a_x - tv_{ix})tv_{ox} - 2t^2v_{oy}v_{iy}]k + [(a_x - tv_{ix})^2 + t^2v_{iy}^2 - D^2] = 0.$$

We solve this equation for  $k$ .

2. Points on the lines  $P'_2$ . We have

$$x = D^2/a_x \quad (4.3)$$

$$y = -\varepsilon D \sqrt{a_x^2 - D^2}/a_x, \quad (4.4)$$

where  $\varepsilon = \pm 1$ . From Equation 4.3,

$$a_x + t(kv_{ox} - v_{ix}) = D^2/a_x$$

$$k = (D^2 - a_x^2)/(ta_x v_{ox}) + v_{ix}/v_{ox}.$$

Equation 4.4 yields to

$$(D^2 - a_x^2)v_{oy}/(a_x v_{ox}) + tv_{ix}v_{oy}/v_{ox} - tv_{iy} = -\varepsilon D \sqrt{a_x^2 - D^2}/a_x. \quad (4.5)$$

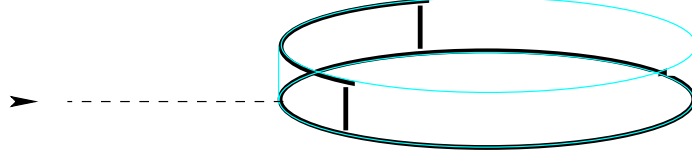


FIG. 3.7. *Target set. Case  $a_z = -H$  and  $-a_x > D$*

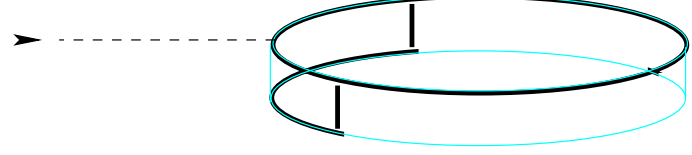


FIG. 3.8. *Target set. Case  $a_z = H$  and  $-a_x > D$*

We solve Equation 4.5 in  $t$ :

$$t = (-\varepsilon D v_{ox} \sqrt{a_x^2 - D^2} + (a_x^2 - D^2) v_{oy}) / (a_x (v_{ix} v_{oy} - v_{ox} v_{iy})).$$

Thus, we deduce

$$k = (v_{iy} \sqrt{a_x^2 - D^2} - v_{ix} \varepsilon D) / (v_{oy} \sqrt{a_x^2 - D^2} - v_{ox} \varepsilon D).$$

We analyze the singularities.

- Case  $v_{oz} = v_{iz}$ . The vector  $\vec{v}_o - \vec{v}_i$  is horizontal and will remain horizontal if we change the ground speed of the ownship. Solutions, if any, are tangential to the circle and also belong to  $P'_2$  and those are handled in case 2. Therefore, there are no solutions.
- Case  $\varepsilon D v_{ox} = v_{oy} \sqrt{a_x^2 - D^2}$ . No solution, since  $t = 0$ .
- Case  $-a_x < D$ . No solution, since the set  $P'_2$  is empty.
- Case  $-a_x = D$ . There is a single line and the relative initial position of the ownship is on that line. The only way to reach a target point on that line is if  $\vec{v}'$  is vertical or null. If  $v_{ix} v_{oy} - v_{ox} v_{iy} = 0$  then there is a solution:  $v'_{ox} = v_{ix}, v'_{oy} = v_{iy}$ . Otherwise, there is none.
- Case  $v_{ix} v_{oy} - v_{ox} v_{iy} = 0$ . The horizontal components of  $\vec{v}_i$  and  $\vec{v}_o$  are parallel, these two horizontal components cannot be the same as the aircraft are predicted to be in conflict and  $-a_x \geq D$ , thus the relative speed  $\vec{v}$  cannot be vertical or null. As the horizontal part of  $\vec{v}_o$  is not 0, the only way to change the direction of the relative speed vector by changing the ownship ground speed is to take  $v'_{ox} = v_{ix}, v'_{oy} = v_{iy}$ . This way,  $\vec{v}'$  is vertical or null. If  $v_{oz} \neq v_{iz}$  then  $\vec{v}'$  is vertical and since  $-a_x > D$ , this is not an optimal solution. If  $v_{oz} = v_{iz}$  then we have the solution  $\vec{v}' = \vec{0}$ .
- Case  $v_{ox} = \vec{0}$ . Since  $v_{ix} v_{oy} - v_{ox} v_{iy} \neq 0$ , we have  $v_{ix} \neq 0$  and  $v_{oy} \neq 0$ . We also have

$$x = a_x - t v_{ix}$$

and  $x$  reaches  $D^2/a_x$  at  $t = (a_x^2 - D^2)/(a_x v_{ix})$  independently of  $k$ . At that time, we must have

$$\begin{aligned} y &= -\varepsilon D \sqrt{a_x^2 - D^2} / a_x \\ (a_x^2 - D^2) / (a_x v_{ix}) (k v_{oy} - v_{iy}) &= -\varepsilon D \sqrt{a_x^2 - D^2} / a_x \\ k &= (v_{iy} - \varepsilon D v_{ix} / (v_{oy} \sqrt{a_x^2 - D^2})). \end{aligned}$$

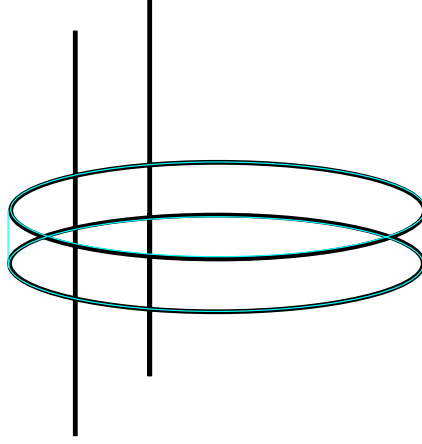


FIG. 4.1. *Superset of target points*

This formula is a particular case of the general one.

3. Special case  $\vec{v}' = \vec{0}$ . The only way to reach  $\vec{v}' = \vec{0}$  by changing the ownship ground speed is to have  $v_{ix}v_{oy} - v_{ox}v_{iy} = 0$  and  $v_{oz} = v_{iz}$ . This is already included as a limit case above.

**4.2. Ground Track Change Only.** Since ground speed and vertical speed are constant, we have

$$v_{ox}'^2 + v_{oy}'^2 = v_{ox}^2 + v_{oy}^2 \quad (4.6)$$

$$v_{oz}' = v_{oz} \quad (4.7)$$

and we must determine the possible  $v_{ox}'$  and  $v_{oy}'$  such that at some time  $t$ , the point  $(x, y, z) = (a_x + t(v_{ox}' - v_{ix}), t(v_{oy}' - v_{iy}), a_z + t(v_{oz}' - v_{iz}))$  is on the target set.

1. Points on the circles  $P_1'$ . We have

$$x^2 + y^2 = D^2 \quad (4.8)$$

$$z = \varepsilon H. \quad (4.9)$$

Equation 4.9 gives  $t = (\varepsilon H - a_z)/(v_{oz} - v_{iz})$ . Equation 4.8 rewrites to

$$\begin{aligned} 2t^2 v_{oy}' v_{iy} &= (a_x - tv_{ix})^2 + t^2 v_{iy}^2 + t^2 v_{ox}'^2 + t^2 v_{oy}'^2 - D^2 + 2(a_x - tv_{ix})tv_{ox}' \\ &= (a_x - tv_{ix})^2 + t^2 v_{iy}^2 + t^2 v_{ox}^2 + t^2 v_{oy}^2 - D^2 + 2(a_x - tv_{ix})tv_{ox}' \\ &= E + 2(a_x - tv_{ix})tv_{ox}', \end{aligned}$$

where  $E = (a_x - tv_{ix})^2 + t^2 v_{iy}^2 + t^2 v_{ox}^2 + t^2 v_{oy}^2 - D^2$ . Therefore,

$$4t^4 v_{oy}'^2 v_{iy}^2 = (E + 2(a_x - tv_{ix})tv_{ox}')^2.$$

Using Equation 4.6, we get

$$4t^4 v_{iy}^2 (v_{ox}^2 + v_{oy}^2 - v_{ox}'^2) = (E + 2(a_x - tv_{ix})tv_{ox}')^2$$

Hence,

$$4t^2 ((a_x - tv_{ix})^2 + t^2 v_{iy}^2) v_{ox}'^2 + 4E(a_x - tv_{ix})tv_{ox}' + E^2 - 4t^4 v_{iy}^2 (v_{ox}^2 + v_{oy}^2) = 0. \quad (4.10)$$

To solve Equation 4.10 on  $v'_{ox}$ , we compute its discriminant  $\Delta = 16v_{iy}^2 t^4 \Delta'$ , where

$$\Delta' = -E^2 + 4t^2(v_{ox}^2 + v_{oy}^2)((a_x - tv_{ix})^2 + v_{iy}^2 t^2).$$

If  $\Delta' \geq 0$ , the solutions are

$$v'_{ox} = (-E(a_x - tv_{ix}) + \varepsilon' v_{iy} t \sqrt{\Delta'}) / (2t((a_x - tv_{ix})^2 + t^2 v_{iy}^2)), \quad (4.11)$$

where  $\varepsilon' = \pm 1$ . Then, we get

$$v'_{oy} = (E + 2(a_x - tv_{ix})tv'_{ox}) / (2t^2 v_{iy}) \quad (4.12)$$

$$= (Etv_{iy} + \varepsilon'(a_x - tv_{ix})\sqrt{\Delta'}) / (2t((a_x - tv_{ix})^2 + t^2 v_{iy}^2)). \quad (4.13)$$

We analyze the singularities.

- Case  $v_{oz} = v_{iz}$ . The vector  $\vec{v}_o - \vec{v}_i$  is horizontal and will remain horizontal if we change the ground track of the ownship. Solutions, if any, are tangential to the circle and also belong to  $P'_2$  and those are handled in case 2.
- Case  $v_{iy} = 0$  and  $a_x \neq tv_{ix}$ . The two solutions of the equation for  $v'_{ox}$  given by the formula 4.11. There are two solutions for  $v'_{oy}$

$$v'_{oy} = \varepsilon' \sqrt{v_{ox}^2 + v_{oy}^2 - v_{ox}^2}.$$

In fact, this formula is a particular case of the general one.

- Case  $v_{iy} = 0$  and  $a_x = tv_{ix}$ . At time  $t$ , the intruder will be where the ownship is at time 0. Changing the ownship ground track does not affect the horizontal distance between the aircraft at time  $t$  (that will be  $t\sqrt{v_{ox}^2 + v_{oy}^2}$  in all cases). Therefore, it does not help to solve the conflict.

2. Points on the lines  $P'_2$ . We have

$$x = D^2 / a_x \quad (4.14)$$

$$y = -\varepsilon D \sqrt{a_x^2 - D^2} / a_x. \quad (4.15)$$

If  $-a_x \neq D$  then  $v'_{ox} = v_{ix}$  is not a solution and Equation 4.14 gives  $t$  in function of  $v'_{ox}$

$$t = \frac{D^2 - a_x^2}{a_x(v'_{ox} - v_{ix})}.$$

Equation 4.15 rewrites to

$$\begin{aligned} (D^2 - a_x^2) / (a_x(v'_{ox} - v_{ix}))(v'_{oy} - v_{iy}) &= -\varepsilon D \sqrt{a_x^2 - D^2} / a_x \\ (v'_{oy} - v_{iy}) / (v'_{ox} - v_{ix}) &= \varepsilon D / \sqrt{a_x^2 - D^2} \\ v'_{oy} &= \varepsilon D / \sqrt{a_x^2 - D^2} (v'_{ox} - v_{ix}) + v_{iy} \end{aligned}$$

Replacing  $v'_{oy}$  in Equation 4.6, we get

$$(\varepsilon D / \sqrt{a_x^2 - D^2} (v'_{ox} - v_{ix}) + v_{iy})^2 + v_{ox}^2 = v_{ox}^2 + v_{oy}^2. \quad (4.16)$$

We solve Equation 4.16 and get  $v'_{ox}$  and then  $v'_{oy}$ .

We analyze the singularities.

- Case  $-a_x < D$ . No solution since the set  $P'_2$  is empty.
- Case  $-a_x = D$ . The only solution is  $v'_{ox} = v_{ix}$ ,  $v'_{oy} = v_{iy}$ . In this case, we must have  $v_{ix}^2 + v_{iy}^2 = v_{ox}^2 + v_{oy}^2$ . Hence,  $\vec{v}$  is vertical or  $\vec{0}$ .

3. Special case  $\vec{v}' = \vec{0}$ . The only way to get  $\vec{v}' = \vec{0}$  by changing the ownship ground track is to have  $v_{ix}^2 + v_{iy}^2 = v_{ox}^2 + v_{oy}^2$  and  $v_{oz} = v_{iz}$ . In this case, we take  $\vec{v}'_o = \vec{v}_i$ .

**4.3. Vertical Speed Change Only.** If the target point is on the line  $P'_2$ , it is also in  $P'_1$ . We only need to consider the cases where the point is in  $P'_1$  and  $\vec{v}' = \vec{0}$ .

1. Points on circles  $P'_1$ . Thus, we have

$$x^2 + y^2 = D^2 \quad (4.17)$$

$$z = \varepsilon H. \quad (4.18)$$

Equation 4.17 rewrites to

$$(a_x + t(v_{ox} - v_{ix}))^2 + (t(v_{oy} - v_{iy}))^2 = D^2. \quad (4.19)$$

We solve Equation 4.19 to get  $t$ . Equation 4.18 yields to

$$v'_{oz} = v_{iz} + (\varepsilon H - a_z)/t.$$

If  $v_{ox} = v_{ix}$ ,  $v_{oy} = v_{iy}$ , Equation 4.19 has a solution only when  $-a_x = D$  and in this case there is no conflict.

2. Special case  $\vec{v}' = \vec{0}$ . The only way to reach  $\vec{v}' = \vec{0}$  by changing the ownship vertical speed is to have  $v_{ox} = v_{ix}$  and  $v_{oy} = v_{iy}$ . We take  $v'_{oz} = v_{iz}$ .

**5. A Prototype Implementation.** We have experimented this algorithm with a prototype implementation. The prototype is about a couple of hundred lines of Java, containing only assignments and conditionals. The functions used are the four operations and square root, but no trigonometric functions (except in the interface, to print the ground track of the aircraft from the computed Cartesian coordinates of the speed vector). The implementation is available at <http://www.icas.edu/~munoz/sources.html>.

Here is a typical execution: We have two aircraft flying at the same altitude with a horizontal separation of 10 nm. In the coordinate system where the intruder is at the origin and the ownship at coordinates  $(-10, 0, 0)$ , the ownship ground track is 0 and the intruder ground track is 180°. The ground speed of the ownship is 400 nm/h and that of the intruder is 300 nm/h. The ownship is climbing at a vertical speed of 1000 ft/mn and the intruder is descending at a vertical speed of -1000 ft/mn. The input to the algorithm is a file containing the following information.

```
Ground distance = 10 nm   Vertical distance = 0 ft
Ownship:      0 deg    400 nm/h   1000 ft/mn
Intruder: 180 deg    300 nm/h   -1000 ft/mn
```

The programs detects a conflict and proposes five solutions:  
Conflict in the time interval (25.7143,29.1456)

There are 5 solutions.

```
Modify GROUND SPEED      317.5889 nm/h   (TOP)
Modify GROUND TRACK      29.1888 deg    (TOP)
Modify GROUND TRACK     -29.1888 deg    (TOP)
Modify VERTICAL SPEED  1266.8799 ft/mn (TOP)
Modify VERTICAL SPEED -3266.8799 ft/mn (BOTTOM)
```

The first solution is to reduce ground speed to 317 nm/h. The second and third modify ground track. The last two solutions modify the vertical speed. On the other hand, in the first four solutions, the target points are on the top circle of the target set. In the last solution, the target point is on the bottom.

Notice that some solutions may not be physically possible. For instance, the last solution proposes an absolute change of vertical speed of more than 4000 ft/mn. The algorithm does not distinguish between the solutions. In fact, it is intended to be used in a more general system, where the choice of *one* solution, among the multiple that have been proposed, may use other kind of information such as type of aircraft, weather conditions, other potential intruders, intent information, etc.

**6. Conclusion.** We have given a complete and rigorous analysis of tactical detection and resolution of air traffic conflicts in the 3-dimensional space and described a new CD&R algorithm that produces a set of solutions. Each solution proposed by the algorithm is a constrained single maneuver that, when performed by the ownship, solves the conflict without collaboration of the intruder aircraft. Experiments have indicated that our algorithm always yields at least two solutions. After thousands of randomly generated examples the average was three solutions per conflict.

Although the algorithm only uses state information, it can be integrated within a more general system, such as AOP, to detect and solve conflicts in piecewise linear flight plans. It is well suited to serve this purpose. First, it is efficient. Particularly, it does not contain loops nor calls to trigonometric functions. Moreover, intent information can be used to chose among the multiple solutions that are proposed. We plan to pursue this direction of research in future work.

In the near future, we will formally verify the correctness of the algorithm in the PVS [10] seek to prove that the solutions proposed indeed solve the conflict and that there is always at least one solution proposed, and extend our approach to deal with multiple aircraft.

**Acknowledgment.** The authors are grateful to Rick Butler for his helpful comments on preliminary versions of this manuscript.

## REFERENCES

- [1] ADVANCED AIR TRANSPORTATION TECHNOLOGIES (AATT) PROJECT, *Concept definition for Distributed Air/Ground Traffic Management (DAG-TM), version 1.0*. NASA Ames Research Center - NASA Langley Research Center, 1999.
- [2] K. BILIMORIA, *A geometric optimization approach to aircraft conflict resolution*, in Guidance, Navigation, and Control Conference, no. AIAA 2000-4265, Denver, CO, August 2000.
- [3] K. BILIMORIA, B. SRIDHAR, G. CHATTERJI, K. SHETH, AND S. GRABBE, *FACET: Future ATM concepts evaluation tool*, in 3rd USA/Europe Air Traffic Management R&D Seminar, Naples, Italy, June 2000.
- [4] V. CARREÑO AND C. MUÑOZ, *Aircraft trajectory modeling and alerting algorithm verification*, in Theorem Proving in Higher Order Logics: 13th International Conference, TPHOLs 2000, J. Harrison and M. Aagaard, eds., vol. 1869 of Lecture Notes in Computer Science, Springer-Verlag, 2000, pp. 90–105. An earlier version appears as report NASA/CR-2000-210097, ICASE No. 2000-16.
- [5] M. EBY, *A self-organizational approach for resolving air traffic conflicts*, Lincoln Laboratory Journal, 7 (1994).
- [6] J. HOEKSTRA, R. RUIGROK, R. VAN GENT, J. VISSER, B. GIJSBERS, M. VALENTI, W. HEESBEEN, B. HILBURN, J. GROENEWEG, AND F. BUSSINK, *Overview of NLR free flight project 1997-1999*, Tech. Report NLR-CR-2000-227, National Aerospace Laboratory (NLR), May 2000.



- [7] J. KUCHAR AND L. YANG, *Survey of conflict detection and resolution modeling methods*, in AIAA Guidance, Navigation, and Control Conference, no. AIAA-97-3732, New Orleans, LA, August 1997, pp. 1388–1397.
- [8] J. McDONALD AND R. VIVONA, *Strategic airborne conflict detection of air traffic and area hazards*, Tech. Report NASA Contract: NAS2-98005 RTO-29, TITAN Systems Corporation, SRC Division, November 2000.
- [9] C. MUÑOZ, R. BUTLER, V. CARREÑO, AND G. DOWEK, *On the verification of conflict detection algorithms*. Submitted as NASA Technical Memorandum. Available at <http://www.icas.edu/~munoz/Drafts/conflict.pdf>, 2001.
- [10] S. OWRE, J. M. RUSHBY, AND N. SHANKAR, *PVS: A prototype verification system*, in 11th International Conference on Automated Deduction (CADE), D. Kapur, ed., vol. 607 of Lecture Notes in Artificial Intelligence, Saratoga, NY, June 1992, Springer-Verlag, pp. 748–752.
- [11] RADIO TECHNICAL COMMISSION FOR AERONAUTICS, *Final report of the RTCA board of directors' select committee on free flight*, Tech. Report Issued 1-18-95, RTCA, Washington, DC, 1995.
- [12] B. SANFORD, K. HARWOOD, S. NOWLIN, H. BERGERON, H. HEINRICHS, G. WELLS, AND M. HART, *Center/TRACON automation system: Development and evaluation in the field*, in 38th Annual Air Traffic Control Association Conference, October 1993.